

## DIAGRAM UNIFIED MODELLING LANGUAGE UNTUK MEMODELKAN LAYANAN AUTOMATED TELLER MACHINE DENGAN PETRI NET

**DORTEUS LODEWYIK RAHAKBAUW**

*Staf Jurusan Matematika FMIPA UNPATTI*

Jl. Ir. M. Putuhena, Kampus Unpatti, Poka-Ambon, Maluku

e-mail: lodewyik@gmail.com

### ABSTRAK

Penelitian ini menguraikan suatu aplikasi yang mungkin dari Petri net untuk menspesifikasikan dinamika dari sistem informasi. Petri net adalah suatu alat matematik yang menerapkan spesifikasi formal dari dinamika sistem. Suatu prosedur yang formal disarankan karena mampu mentransformasi diagram kegiatan *Unified Modeling Language* (UML) ke dalam suatu model Petri net. Atas dasar perubahan bentuk ini dimungkinkan untuk memenuhi verifikasi model dinamis dari sistem riil, yaitu untuk mengevaluasi apakah aktivitas dan ordernya terdefinisi dengan baik (*well defined*). Ini juga mungkin untuk memecahkan permasalahan alur dan sinkronisasi aktivitas suatu sistem, seperti juga untuk mengoptimalkan model dinamis. Permasalahan yang dibahas dalam tulisan ini adalah bagaimana memodelkan layanan ATM dengan berdasar pada diagram aktivasi UML (*Unified Modelling Language*) dan bagaimana memodelkan layanan ATM (*Automated Teller Machine*) dengan menggunakan Petri Nets. Dihasilkan 49 Transisi dan 26 place untuk menggambarkan kedinamikan suatu ATM.

**Kata kunci :** *Petri Net, ATM, UML*

### PENDAHULUAN

Petri net sebagai satu alat pelengkap untuk model aktivitas dari suatu sistem. Aktivitas digambarkan dalam satu diagram kegiatan UML (*Unified Modeling Language*) yaitu suatu diagram kegiatan menunjukkan interaksi antara objek, dalam kaitan dengan menggunakan istilah aktivitas. Aktivitas diwakili sebagai *state action* dan transisi-transisi antara state secara implisit dicetuskan oleh penyelesaian tindakan-tindakan di dalam sumber state.

Proses pengembangan dari suatu sistem informasi meliputi spesifikasi yang statis dan struktur yang dinamis dari suatu sistem. Dalam beberapa tahun terakhir suatu pendekatan berorientasi objek mempunyai dominan dalam pengembangan sistem informasi. Pendekatan itu didasarkan pada fakta bahwa objek dan hubungan-hubungan tersebut mewakili karakteristik-karakteristik yang riil dari suatu sistem dalam suatu pengembangan. Sistem itulah yang menghubungkan suatu objek dengan

objek yang lain dan saling menghubungkan. Tiap-tiap state dari sistem didefinisikan oleh state-state dari objek.

### TINJAUAN PUSTAKA

#### PETRI NET

Petri net adalah suatu alat pemodelan matematik secara grafis. Dikembangkan pertama kali oleh C.A. Petri pada tahun 1962 [1], yang terdiri dari *place-place*, *transisi-transisi*, dan anak panah yang menghubungkan *place* dan *transisi*. Arah masukan anak panah (*arc*) menghubungkan suatu *input place* ke *transisi* dan jika arah masuk anak panah bergerak dari suatu *transisi* maka akan berakhir pada *output place*. *place* dapat diisi dengan beberapa *token*. keadaan pada suatu sistem pemodelan ditandai dengan penomoran/tanda (tipe dari tiap *token* dapat dibedakan) pada setiap *place*. *Transisi-transisi* merupakan komponen aktif. Model aktivitas tersebut dapat terjadi ketika (*transisi fires*), kemudian mengubah keadaan dari satu sistem (penandaan dari Petri net). *Transisi-transisi*

hanya dapat dikatakan *fires* jika berstatus *enabled*, (dalam artian setiap *place* mempunyai cukup *token*). Ketika *transisi* menembak, *token* pada *input place* akan berkurang dan ditambahkan pada *output place* yang dituju. Jumlah token yang berpindah bergantung pada bobot (*weight*) pada tiap-tiap *arc*.

Petri net adalah 4-tuple  $(P, T, A, w)$  dengan

- $P$  : himpunan berhingga place,  $P = \{p_1, p_2, \dots, p_n\}$ ,
- $T$  : himpunan berhingga transisi,  $T = \{t_1, t_2, \dots, t_m\}$ ,
- $A$  : himpunan arc,  $A \subseteq (P \times T) \cup (T \times P)$
- $w$  : fungsi bobot,  $w : A \rightarrow \{1, 2, 3, \dots\}$

maka himpunan place dan transisi tidak harus berupa himpunan berhingga melainkan bisa berupa himpunan takhingga.

## HASIL DAN PEMBAHASAN

### DIAGRAM AKTIVITAS

Diagram aktivitas adalah sesuatu yang khusus dalam diagram *statechart* dari UML di mana semua *state* adalah suatu tindakan, dan transisi-transisi itu tentukan oleh penyelesaian tindakan-tindakan dalam sumber *state*. Diagram aktivitas berhubungan dengan suatu kelas, untuk mengimplementasikan suatu operasi yang terjadi pada kelas/suatu kasus. Tujuan dari diagram adalah ini difokuskan pada alur suatu control dan pembawa data oleh proses internal.

*State action* adalah suatu keadaan dengan tindakan internal dan sedikitnya satu transisi yang berperan dalam setiap tindakan internal. Jika ada beberapa transisi yang dapat digunakan maka transisi-transisi tersebut harus mempunyai kondisi bersyarat. *state action* digunakan untuk memodelkan setiap langkah dalam tiap eksekusi dari suatu algoritma atau prosedur. Setiap keputusan menyatakan situasi ketika kondisi bersyarat digunakan untuk menandai ada tidaknya transisi-transisi yang mungkin. Transisi-transisi tersebut disebut transisi keluaran, dan transisi-transisi lain disebut transisi masukan.

Perubahan bentuk tiap-tiap sesi dari diagram aktivitas *Unified Modeling Language* ke dalam suatu model Petri net perlu dipertimbangkan. Perubahan bentuk *state* aktivitas, transisi-transisi dan keputusan-keputusan yang internal digambarkan dengan cara yang formal. Dengan demikian suatu definisi dari diagram aktivitas serta struktur dari petri net dapat terjawab.

Lebih lanjut, suatu transisi yang kompleks mungkin mempunyai sumber *state* aktivitas dan targetnya. Hal ini merepresentasi sinkronisasi dan/ atau pemisahan kendali ke dalam alur-alur secara berbarengan. Suatu transisi kompleks dikatakan *enabled* ketika semua *state* dapat terlewati. Transisi kompleks yang menunjukkan sinkronisasi disebut *join*, dan yang menunjukkan pemisahan disebut *fork*.

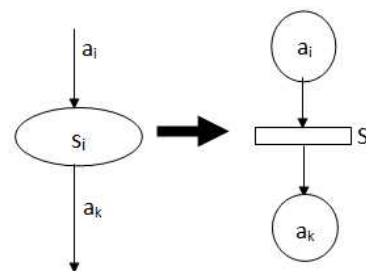
### MENTRANSFORMASIKAN STATE AKTIVITAS DAN TRANSISI-TRANSISI INTERNAL KE DALAM SUATU MODEL PETRI NET.

Perubahan bentuk diagram aktivitas ke dalam suatu model Petri net didasarkan pada perubahan bentuk perintah yang

ditunjukkan dalam gambar 1-3. Gambar 1 menunjukkan aturan berhubungan dengan perubahan bentuk dari suatu *state* aktivitas dan menyertakan transisi-transisi masukan dan keluaran yang internal. *State* aktivitas  $s_i$  diubah menjadi  $s_i$  transisi dari suatu Petri net. Suatu transisi masukan  $a_i$  dari *state* aktivitas  $s_i$  diubah menjadi *place* masukan  $a_i$  dari  $s_i$  transisi diubah menjadi tempat masukan  $a_i$  dari transisi  $s_i$  suatu Petri net. Transisi keluaran  $a_k$  dari *state* aktivitas  $s_i$  diubah menjadi tempat keluaran  $a_k$  dari transisi  $s_i$  dari suatu Petri net.

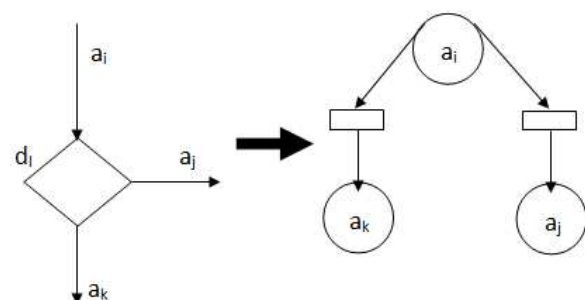
Perubahan bentuk suatu keputusan dan menyertakan transisi-transisi masukan dan keluaran dari diagram aktivitas ke dalam *place-place* dan transisi-transisi suatu Petri net ditunjukkan di dalam gambar 2. Transisi masukan  $a_i$  diubah menjadi *place* masukan  $a_i$  dan transisi-transisi keluaran  $a_j$  dan  $a_k$  diubah menjadi *place* keluaran  $a_k$  dan  $a_j$  dari suatu Petri net. Suatu keputusan ditunjukkan dengan dua atau lebih transisi. Banyaknya transisi-transisi pada suatu Petri net sama banyaknya dengan transisi-transisi keluaran pada suatu keputusan.

Diagram aktivitas dengan transisi keluaran lebih dari satu dapat diperagakan seperti yang ditunjukkan di dalam gambar 3. *State* Aktivitas menyatakan  $s_i$  dimodelkan dengan



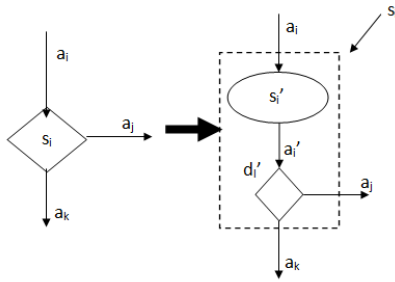
**Gambar 1:** Perubahan bentuk perintah untuk satu *state* aktivitas

*state* aktivitas  $s_i'$  dan keputusan  $d_i'$ , yang dihubungkan dengan transisi  $a_i'$ . transisi masukan  $a_i$  dari *state* aktivitas  $s_i$  menjadi transisi masukan dari *state* aktivitas  $s_i'$ .



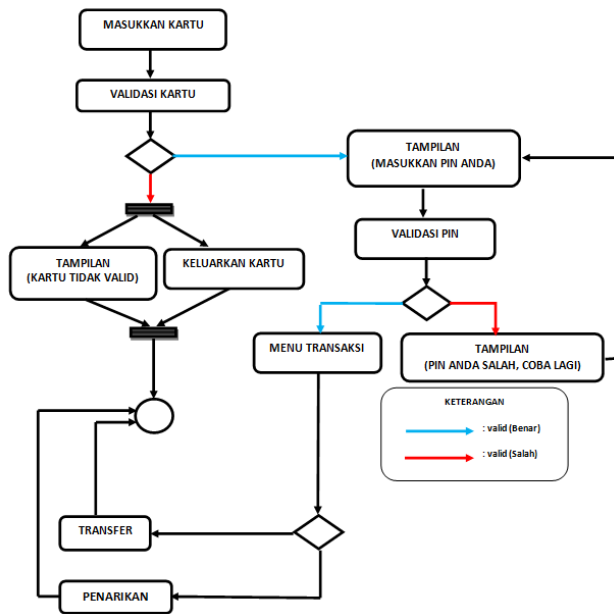
**Gambar 2:** Perubahan bentuk perintah untuk suatu keputusan

transisi-transisi keluaran  $a_j$  dan  $a_k$  menjadi transisi-transisi keluaran keputusan  $d_i'$ . struktur ini dapat diubah menjadi suatu Petri net menurut aturan menggambar di atas yang ditunjukkan oleh gambar-gambar 1 dan 2.

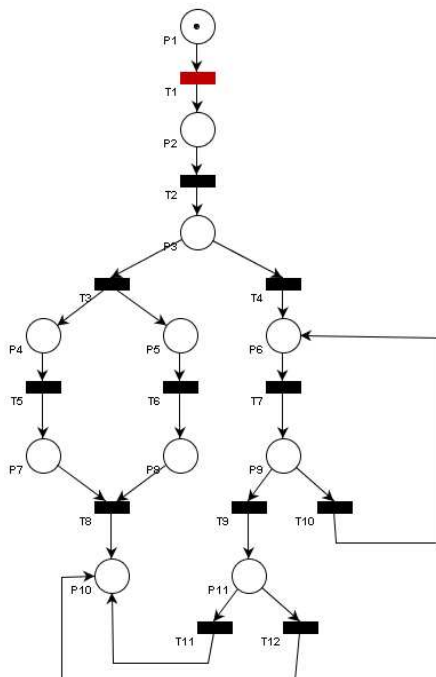


**Gambar 3:** Perubahan bentuk memerintah karena *state* aktivitas dengan sisa transisi lebih dari satu

### CONTOH ILUSTRASI



**Gambar 4 :** Diagram aktivitas untuk mesin ATM



**Gambar 5 :** Model Petri net dari Diagram aktivitas pada Gambar 4

### 3 kasus.

1. Kartu valid PIN valid
2. Kartu tidak valid
3. Kartu valid PIN tidak valid

### Matriks *forward incidence*.

0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	0	0	0	0	0
0	1	0	0	0	0	0	0	0	0	0	0	0
0	0	1	0	0	0	0	0	0	0	0	0	0
0	0	1	0	0	0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	0	0	1	0	0	0
0	0	0	0	1	0	0	0	0	0	0	0	0
0	0	0	0	0	1	0	0	0	0	0	0	0
0	0	0	0	0	0	1	0	0	0	0	0	0
0	0	0	0	0	0	0	1	0	0	1	1	0
0	0	0	0	0	0	0	0	1	0	0	0	0

### Matriks *Backward incidence*.

1	0	0	0	0	0	0	0	0	0	0	0	0
0	1	0	0	0	0	0	0	0	0	0	0	0
0	0	1	1	0	0	0	0	0	0	0	0	0
0	0	0	0	1	0	0	0	0	0	0	0	0
0	0	0	0	0	1	0	0	0	0	0	0	0
0	0	0	0	0	0	1	0	0	0	0	0	0
0	0	0	0	0	0	0	1	0	0	0	0	0
0	0	0	0	0	0	0	0	1	0	0	0	0
0	0	0	0	0	0	0	0	0	1	1	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	1	1

Kadaan awal petri net

$$X_0 = [1 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0]^T$$

### 1. Kartu Valid Pin Valid

$$X_1 = [0 \ 1 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0]^T$$

$$X_2 = [0 \ 0 \ 1 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0]^T$$

$$X_3 = [0 \ 0 \ 0 \ 0 \ 0 \ 1 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0]^T$$

$$X_4 = [0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 1 \ 0 \ 0 \ 0 \ 0]^T$$

$$X_5 = [0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 1 \ 0]^T$$

$$X_6 = [0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 1 \ 0 \ 0 \ 0]^T$$

Dengan susunan transisi yang di fire secara berurutan adalah :  $T_1, T_2, T_4, T_7, T_9, T_{10}$

## 2. Kartu tidak valid

$$X_1 = [0\ 1\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0]^T$$

$$X_2 = [0\ 0\ 1\ 0\ 0\ 0\ 0\ 0\ 0\ 0]^T$$

$$X_3 = [0\ 0\ 0\ 1\ 1\ 0\ 0\ 0\ 0\ 0]^T$$

$$X_4 = [0\ 0\ 0\ 0\ 1\ 0\ 1\ 0\ 0\ 0]^T$$

$$X_5 = [0\ 0\ 0\ 0\ 0\ 0\ 1\ 1\ 0\ 0]^T$$

$$X_6 = [0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 1]^T$$

Dengan susunan transisi yang di fire secara berurutan adalah :  $T_1, T_2, T_3, T_5, T_6, T_8$

## 3. Kartu valid PIN tidak valid

$$X_1 = [0\ 1\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0]^T$$

$$X_2 = [0\ 0\ 1\ 0\ 0\ 0\ 0\ 0\ 0\ 0]^T$$

$$X_3 = [0\ 0\ 0\ 0\ 0\ 1\ 0\ 0\ 0\ 0]^T$$

$$X_4 = [0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 1\ 0]^T$$

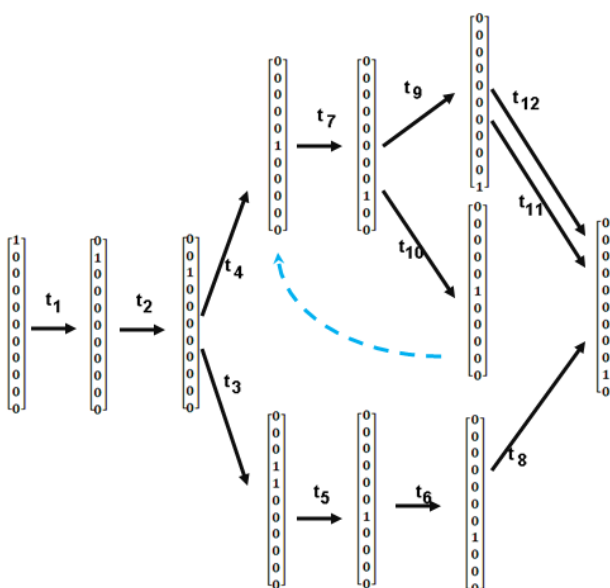
$$X_5 = [0\ 0\ 0\ 0\ 0\ 1\ 0\ 0\ 0\ 0]^T$$

$$X_6 = [0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 1\ 0]^T$$

Ket: Yang di tandai kurung kurawal berulang.

Dengan susunan transisi yang di fire secara berurutan adalah :  $T_1, T_2, T_4, T_7, T_{10}, T_7, T_{10}, \dots$

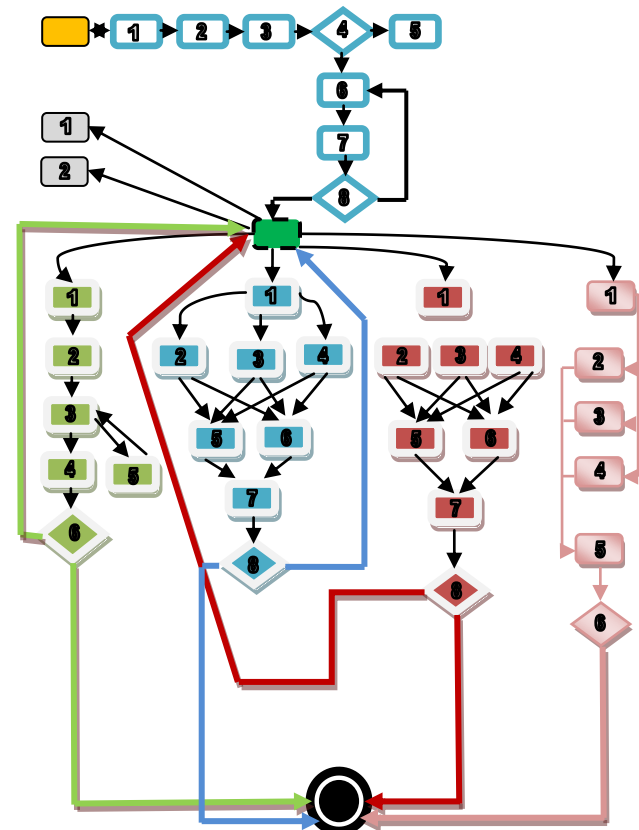
## Coverability tree



Tahap pertama yang dilakukan dalam membangun *coverability tree* adalah menentukan *node root*. Node root menyatakan keadaan awal Petri net yaitu  $[1\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0]^T$ . Pada keadaan ini transisi  $t_1$  *enabled* dengan memfire transisi ini keadaan Petri net

menjadi  $[0\ 1\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0]^T$  transisi yang *enabled* pada keadaan ini adalah  $t_2$ , dan keadaan Petri net menjadi  $[0\ 0\ 1\ 0\ 0\ 0\ 0\ 0\ 0\ 0]^T$  setelah  $t_2$  di *fire*. Berikutnya ada dua transisi yang *enabled* yaitu  $t_3$  dan  $t_4$  dan apabila  $t_3$  di *fire* maka keadaan Petri net menjadi  $[0\ 0\ 0\ 1\ 1\ 0\ 0\ 0\ 0\ 0]^T$ , pada keadaan ini  $t_5$  dan  $t_6$  *enabled* dan bila dilakukan pemfirean berturut turut  $t_5, t_6$  maka didapat juga keadaan Petri net secara berturut-turut adalah  $[0\ 0\ 0\ 0\ 1\ 0\ 1\ 0\ 0\ 0]^T, [0\ 0\ 0\ 0\ 0\ 1\ 1\ 0\ 0]^T$ , sekarang pada keadaan Petri net  $[0\ 0\ 0\ 0\ 0\ 1\ 1\ 0\ 0]^T$  transisi yang *enabled* hanya  $t_8$  dan bila di lakukan pemfirean didapat keadaan akhir Petri net adalah  $[0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 1\ 0]^T$ . Selanjutnya pada saat keadaan Petri net  $[0\ 0\ 1\ 0\ 0\ 0\ 0\ 0\ 0\ 0]^T$ , ada dua transisi yang *enabled* yaitu  $t_3$  dan  $t_4$ . Sudah dilakukan pemfirean untuk  $t_3$  dan apabila dilakukan pemfirean untuk  $t_4$  akan merubah keadaan Petri net menjadi  $[0\ 0\ 0\ 0\ 0\ 1\ 0\ 0\ 0\ 0]^T$ , sekarang satu-satunya transisi yang *enabled* adalah  $t_7$  dan keadaan Petri net menjadi  $[0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 1\ 0]^T$  setelah  $t_7$  di *fire*. Sekarang ada  $t_9$  dan  $t_{10}$  yang *enabled* dan apabila  $t_{10}$  di *fire* akan kembali ke keadaan Petri net saat  $[0\ 0\ 0\ 0\ 0\ 1\ 0\ 0\ 0\ 0]^T$ , sedangkan apabila  $t_9$  yang di *fire* maka akan merubah keadaan Petri net menjadi  $[0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 1]^T$  sekarang ada dua transisi yang *enabled* yaitu  $t_{11}$  dan  $t_{12}$  dan walaupun salah satunya di *fire* tetap akan merubah keadaan akhir Petri net menjadi  $[0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 1\ 0]^T$ .

## SIMULASI LAYANAN PADA ATM (AUTOMATED TELLER MACHINE) BNI



Gambar 6 : Diagram aktivitas untuk mesin ATM BNI

**Tabel 1.**

Keterangan Transisi dari petri net pada Gambar 6

No	Kode Transisi	Keterangan
1	T <sub>0</sub>	Nasabah masuk ke ATM
2	T <sub>1</sub>	Nasabah memasukkan kartu ATM
3	T <sub>2</sub>	Validasi kartu
4	T <sub>3</sub>	Kartu tidak valid
5	T <sub>4</sub>	Kartu valid
6	T <sub>5</sub>	Nasabah memasukkan PIN
7	T <sub>6</sub>	Validasi PIN
8	T <sub>7</sub>	PIN tidak valid
9	T <sub>8</sub>	PIN valid
10	T <sub>9</sub>	<b>GANTI PIN</b>
11	T <sub>10</sub>	Registrasi PIN lama
12	T <sub>11</sub>	Registrasi PIN baru
13	T <sub>12</sub>	Format PIN salah
14	T <sub>13</sub>	Pembatalan Ganti PIN
15	T <sub>14</sub>	Stop transaksi
16	T <sub>15</sub>	Transaksi lain
17	T <sub>16</sub>	<b>TRANSFER</b>
18	T <sub>17</sub>	Pembatalan transfer
19	T <sub>18</sub>	Dari rekening giro
20	T <sub>19</sub>	Dari rekening tabungan
21	T <sub>20</sub>	Dari kartu kredit
22	T <sub>21</sub>	Ke rekening bank lain
23	T <sub>22</sub>	Ke rekening BNI
24	T <sub>23</sub>	Masukkan no rekening bank lain
25	T <sub>24</sub>	Masukkan no rekening BNI
26	T <sub>25</sub>	Stop transaksi
27	T <sub>26</sub>	Transaksi lain
28	T <sub>27</sub>	<b>PEMBAYARAN</b>
29	T <sub>28</sub>	<b>REGISTRASI e-CHANEL</b>
30	T <sub>29</sub>	<b>PENARIKKAN TUNAI</b>
31	T <sub>30</sub>	Dari rekening giro
32	T <sub>31</sub>	Dari rekening tabungan
33	T <sub>32</sub>	Dari kartu kredit
34	T <sub>33</sub>	Lainnya
35	T <sub>34</sub>	Salah masukkan nominal
36	T <sub>35</sub>	Masukkan nominal sudah benar
37	T <sub>36</sub>	Nominal 250.000
38	T <sub>37</sub>	Nominal 500.000
39	T <sub>38</sub>	Nominal 1.000.000
40	T <sub>39</sub>	Nominal 1.200.000
41	T <sub>40</sub>	Pengambilan dengan bukti struk
42	T <sub>41</sub>	Pengambilan tanpa bukti struk
43	T <sub>42</sub>	<b>INFORMASI SALDO</b>
44	T <sub>43</sub>	Rekening giro
45	T <sub>44</sub>	Rekening tabungan
46	T <sub>45</sub>	Kartu kredit
47	T <sub>46</sub>	Transaksi lain
48	T <sub>47</sub>	Stop transaksi
49	T <sub>48</sub>	Pengambilan kartu
50	T <sub>49</sub>	Pengambilan struk

**Tabel 2.**

Keterangan Place dari petri net pada Gambar 6

No	Kode Transisi	Keterangan
1	P <sub>0</sub>	Antrian nasabah
2	P <sub>1</sub>	Nasabah didalam ATM
3	P <sub>2</sub>	Kartu masuk ATM
4	P <sub>3</sub>	Proses validasi kartu ATM
5	P <sub>4</sub>	Kartu yang tidak valid di tampung di ATM
6	P <sub>5</sub>	Kartu valid
7	P <sub>6</sub>	Menu masukkan PIN
8	P <sub>7</sub>	Proses validasi PIN
9	P <sub>8</sub>	Tampilan menu utama
10	P <sub>9</sub>	Masukkan PIN lama
11	P <sub>10</sub>	Masukkan PIN baru
12	P <sub>11</sub>	Menu mau lanjut transaksi atau tidak
13	P <sub>12</sub>	Menu tampilan Transfer
14	P <sub>13</sub>	Pilih transfer ke Rek.BNI / bank lain
15	P <sub>14</sub>	Masukkan no rek bank lain
16	P <sub>15</sub>	Masukkan no rek BNI
17	P <sub>16</sub>	Menu mau lanjut transaksi atau tidak
18	P <sub>17</sub>	<b>Menu PEMBAYARAN</b>
19	P <sub>18</sub>	<b>Menu Reg. e-channel</b>
20	P <sub>19</sub>	Menu transaksi arah penarikan
21	P <sub>20</sub>	Nominal transaksi penarikan
22	P <sub>21</sub>	Input Nominal yang tidak ada pada menu
23	P <sub>22</sub>	Mau cetak receipt atau tidak
24	P <sub>23</sub>	Proses cetak receipt
25	P <sub>24</sub>	Menu arah informasi saldo
26	P <sub>25</sub>	Jumlah Saldo, mau Transaksi lain
27	P <sub>26</sub>	Terima kasih sudah melakukan transaksi

**KESIMPULAN**

1. Petri net mampu merepresentasikan cara kerja mesin ATM dengan 49 Transisi dan 26 Place.
2. Dalam pengembangan sistem yang lebih besar sekalipun seperti ATM Petri net mampu merepresentasikan selama sistem bersifat tidak *deadlock*

**DAFTAR PUSTAKA**

- Adzkiya, D. 2008, *Membangun Petri Net Lampu Lalu Lintas dan Simulasinya*, Tesis Magister, Institut Teknologi Sepuluh Nopember, Surabaya.
- Bordbar, B., Giacomini, L., Holding, D.J., Design Of Distributed Manufacturing Systems Using Uml And Petri Nets, Department of Electronic Engineering, School of Engineering, Aston University, Aston Triangle, Birmingham.

- David, R. dan Alla, H. 2005, *Discrete, Continuous, and Hybrid Petri Nets*, Springer-Verlag Berlin Heidelberg, New York.
- Saldhana, J.A., Shatz, S.M., UML Diagrams to Object Petri Net Models: An Approach for Modeling and Analysis, Department of Electrical Engineering and Computer Science University of Illinois, Chicago.
- Storrie, H, Models of Software Architecture, Fakultät für Mathematik und Informatik Ludwig-Maximilians-Universität München